AD-779 142

AN ADVANCED DUAL BASIC FEASIBLE SOLUTION FOR A CLASS OF CAPACITATED GENERALIZED NETWORKS

John Hultz, et al

Texas University

Prepared for:

Office of Naval Research

April 1974

Unclassified
Security Classification

AD 779 142

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* Center for Cybernetic Studies The University of Texas	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE
An Advanced Dual Basic Feasible Solution For A Class Of Capacitated Generalized Networks

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

5. AUTHOR(S) *(First name, middle initial, last name)*
John Hultz
Darwin Klingman
Robert Russell

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
April 1974	19	22

8a. CONTRACT OR GRANT NO. N00014-67-A-0126-0008; 0009 b. PROJECT NO. NR 047-021 c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) Center for Cybernetic Studies Research Report CS 170 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*

10. DISTRIBUTION STATEMENT
This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Office of Naval Research (Code 434) Washington, D. C.

13. ABSTRACT

This paper presents a "one pass" algorithm that determines an "advanced" dual basic feasible solution for a class of capacitated generalized network problems. Special cases in this class of problems include transportation and transshipment problems. Computational results are included which show that this new start substantially improves the solution performance of the dual method for transportation and transshipment problems. In fact, a dual code employing this advanced start is found to be faster (in terms of total solution time) than the fastest out-of-kilter code SUPERK on highly rectangular transportation problems.

I

DD FORM 1 NOV 65 1473 (PAGE 1)
S/N 0101-807-6811

Unclassified
Security Classification

A-31408

Unclassified
Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Networks						
Computational Testing						
Linear Programming						
Transportation						
Minimum Cost Flow						
Generalized Networks						

II

DD FORM 1 NOV 65 1473 (BACK)
S/N 0102-014-6800

Unclassified
Security Classification

A-31409

Research Report
CS 170

AN ADVANCED DUAL BASIC FEASIBLE SOLUTION FOR A CLASS OF CAPACITATED GENERALIZED NETWORKS

by

John Hultz
Darwin Klingman
Robert Russell*

April 1974

*Assistant Professor of Management, University of Tulsa, Tulsa, Oklahoma.

This research was partly supported by ONR Project NR 047-021, Contracts N00014-67-A-0126-0008 and N00014-67-A-0126-0009 with the Center for Cybernetic Studies, The University of Texas. Reproduction in whole or in part is permitted for any purpose of the United States Government.

CENTER FOR CYBERNETIC STUDIES

A. Charnes, Director
Business-Economics Building, 512
The University of Texas
Austin, Texas 78712

ABSTRACT

This paper presents a "one pass" algorithm that determines an "advanced" dual basic feasible solution for a class of capacitated generalized network problems. Special cases in this class of problems include transportation and transshipment problems. Computational results are included which show that this new start substantially improves the solution performance of the dual method for transportation and transshipment problems. In fact, a dual code employing this advanced start is found to be faster (in terms of total solution time) than the fastest out-of-kilter code SUPERK on highly rectangular transportation problems.

1. Introduction

This paper presents a "one pass" algorithm that determines an "advanced" dual basic feasible solution for a class of capacitated generalized network problems. Special cases in this class of problems include the standard transportation problem and the pure network or transshipment problem.

The algorithm presented is "advanced" in that it determines a solution which is not only feasible but relatively close to the optimum of the dual of a generalized network problem. The advanced algorithm is a direct extension of the dual start algorithm of Glover, Klingman and Napier [13]. Their algorithm was the first to obtain dual feasible solutions for a class of multi-source, multi-sink, capacitated generalized network problems, and extended the one pass algorithm of Charnes and Raike [4] for determining an optimal solution for another class of generalized network problems The advanced algorithm of this paper differs from the earlier dual start [13] in that objective function information is used in constructing the dual feasible solution.

This research was motivated by recent computational investigations [1, 9, 11, 12, 17, 19] which have found that special dual simplex approaches to transshipment and transportation problems are generally inferior to special purpose primal simplex and primal dual (or out-of-kilter) approaches on large problems. In particular these studies have found that the special purpose primal simplex approaches [9,11,17] are substantially (4-10 times) faster than out-of-kilter approaches and that the out-of-kilter codes are faster than dual codes which employ the dual start in [13]. Since the study [1] showed that it is possible to increase the efficiency of the out-of-kilter method by at least a factor of four, we felt that further computational investigation on dual approaches was warranted. Furthermore, since the reported computational testing on the dual approach [9, 12] indicated that its major computational weakness lies in the large time required to execute a pivot, it seemed reasonable that starting procedures which would reduce the number of pivots required to find an optimal solution should be designed and computationally tested.

Computational investigation on the advanced start developed herein shows that it acheives this reduction. In particular, the results indicate that the advanced start reduces both the solution time and number of pivots over other dual approaches using the start in [13] by roughly 50%.

While this result still is not able to make the dual approach competitive with the primal approach on most transportation and transshipment problems, our computational testing does indicate that for highly rectangular transportation problems (i.e., transportation problems with only a few origins and a large number of destinations) the dual approach using the advanced start procedure is superior to the fastest known out-of-kilter code [1] and competitive with the fastest primal codes [9, 11, 17].

A principal motivation cited by other papers [3, 20] for applying the dual method to network problems is the possibility that the problem's 'supplies' and 'demands' may not be permanently fixed, but rather are subject to change. In such a situation, the ability to begin from an optimal basis to a given problem and proceed via the dual method to an optimal solution for the altered problem is extremely useful. Our computational results indicate that the advanced start procedure, however, offers a new motivation for the use of the dual method--namely, by use of our algorithm, it is possible to employ the dual method not only for 'postoptimization,' but also for solving the original problem directly.

2. Problem Statement

A transshipment problem will be defined as consisting of m nodes that are connected pairwise by a collection of n directed arcs. Let N denote the set of all admissible arcs (i,j) in a transshipment problem, then a capacitated generalized transshipment problem can be stated mathematically as:

$$\text{minimize} \quad \sum_{(i,j)\in N} c_{ij}\, x_{ij}$$

$$\text{subject to} \quad -\sum_{(i,j)\in N} x_{ij} + \sum_{(j,i)\in N} k_{ji} x_{j} = b_i, \quad (i = 1,2,\ldots,m) \qquad (1)$$

$$L_{ij} \leqq x_{ij} \leqq u_{ij}, \quad [(i,j)\in N]$$

where

1. x_{ij} is the flow from node i to node j.
2. c_{ij} is the cost of sending a single unit of flow from node i to node j.
3. b_i is the amount of supply (demand) at node i, where the supply (demand) at node i is denoted by a negative (positive) b_i.
4. k_{ij} is the attenuation factor affecting the amount of flow leaving the initial node i and flowing into terminal node j.

The dual to Problem (1) may be stated as:

$$\text{maximize} \quad \sum_{i=1}^{m} b_i w_i + \sum_{(i,j)\in N} s_{ij} L_{ij} - \sum_{(i,j)\in N} t_{ij} U_{ij}$$

$$\text{subject to} \quad k_{ij} w_j - w_i + s_{ij} - t_{ij} = c_{ij} \; ; \quad [(i,j) \in N] \qquad (2)$$

$$w_i \text{ unrestricted} \; ; \quad (i = 1,2,\ldots,m)$$

$$s_{ij}\; t_{ij} \geq 0 \qquad [(i,j) \in N]$$

The advanced algorithm presented in the next section finds a basic feasible solution (if it exists) for Problem (2) provided (a) $c_{ij} \geq 0$, $(i,j) \in N$, (b) $0 < k_{ij} \leq 1$, $(i,j) \in N$, and (c) the rank of the incidence matrix is $m - 1$.

Conditions (a) - (c) are not as restrictive as they may appear. Requirement (a) is generally innocuous in operations research applications [22, p. 178]. Restriction (b) prohibits the inclusion of a loop in the dual feasible basis determined by the algorithm. Condition (c) is always satisfied by transportation and pure network problems; it is only restrictive in the generalized network case when the rank of the incidence matrix is m. Even in this case, the algorithm will generate a dual feasible spanning tree solution, although not a basic solution. Requirement (c) can be circumvented if the generalized network contains a single master source with an unlimited supply.

In order to generate a dual basic feasible solution to Problem (2), the algorithm first derives a basic feasible solution to the problem:

$$\text{maximize} \quad \sum_{i=1}^{m} b_i w_i$$

$$\text{subject to} \quad k_{ij} w_j - w_i \leq c_{ij} \qquad [(i,j) \in N] \qquad (3)$$

$$w_i \quad \text{unrestricted}$$

Problem (3) is the dual to Problem (1) in uncapacitated form.

The algorithm presented in the next section will generate a basic feasible solution to Problem (3). Clearly such a solution will yield a feasible solution to Problem (2). The algorithm proceeds by first selecting an appropriate starting node and determining the generalized

longest path from all other nodes that can reach the starting node by a directed arc. One of the above nodes is then selected (and its associated dual variable value fixed) in such a manner that dual feasibility is preserved and the objective function (at this iteration) is maximized. At each subsequent stage generalized shortest and longest paths are determined from fixed to non-fixed nodes; one node is again selected whose dual variable value preserves feasibility and yields the best return for the objective function. The process is repeated until all m nodes of the network have been selected. The algorithm is a "one-pass" method in the sense of Charnes and Raike [4] in that at each interation one new node is selected; the process then yields a succession of m single steps which determine the solution.

3. Algorithm

Let

$T(p)$ = the set of "fixed" nodes at iteration p.

S = the set of arcs from fixed nodes to nonfixed nodes.

R = the set of arcs from nonfixed nodes to fixed nodes

B = the set of arcs currently selected to be in the basis

1. Select the starting node t to be a source node. Set the dual variable, w_t, of this node equal $\overline{M} =$

$$\frac{\sum_{(i,j)\in N} c_{ij}}{\prod_{(i,j)\in N} k_{ij}}$$

(For the pure network problem, any node can be selected arbitrarily as the starting node t and $\overline{M}$ can be set equal to zero.) Initialize by setting $T(1) = \{t\}$ and $B = \emptyset$. Set $q = t$; in general q denotes the node whose dual variable was last set. Set $p = 1$; in general p denotes the current iteration of the algorithm. Set $w_i^1(0) = \infty$ for $i \notin T(1)$ and $w_i^2(0) = 0$ for $i \notin T(1)$.

2. If $S = \{(i,j)\in N: \; i\in T(p), \; j\notin T(p)\}$ is empty, proceed to step 3; otherwise for each node $j\notin T(p)$, let

$$w_j^1(p) = \begin{cases} \min\{w_j^1(p-1), \dfrac{w_q + c_{qj}}{k_{qj}}\} & \text{if } (q,j) \in N \\ w_j^1 & \text{if } (q,j) \notin N \end{cases}$$

3. If $R = \{(i,j) \in N: i \notin T(p), j \in T(p)\}$ is empty, proceed to step 4; otherwise for each node $j \notin T(p)$, let

$$w_j^2(p) = \begin{cases} \max\{w_j^2(p-1), k_{jq}w_q - c_{jq}\} & \text{if } (j,q) \in N \\ w_j^2(p-1) & \text{if } (i,q) \notin N \end{cases}$$

4. If $S \cup R$ is empty and there exists a node $j \notin T(p)$, then stop for the network is disconnected; otherwise if $S \neq \emptyset$ set $w_f = \min_{j \notin T(p)} \{w_j^1(p)\}$ (where f equals the $j \notin T(p)$ yielding the minimum) and if $R \neq \emptyset$ set $w_r = \max_{j \notin T(p)} \{w_j^2(p)\}$ (where r equals the $j \notin T(p)$ yielding the maximum).

Next compute the "effective net stock" for node f (if $S \neq \emptyset$) as

$$b_f^1 = b_f + \sum_{\substack{(f,j) \in N \\ j \notin T(p)}} b_j/k_{fj} + \sum_{\substack{(j,f) \in N \\ j \notin T(p)}} k_{jf}b_j$$

and if $R \neq \emptyset$ compute

$$b_r^1 = b_r + \sum_{\substack{(r,j) \in N \\ j \notin T(p)}} b_j/k_{rj} + \sum_{\substack{(j,r) \in N \\ j \notin T(p)}} k_{jr}b_j$$

Then if $w_f b_f^1 \geq w_r b_r^1$, set $q = f$ and augment T(p) by q; otherwise set $q = r$ and augment T(p) by q. If $q = f$ augment B by the arc (p,f) for which $w_f = \dfrac{w_p + c_{pf}}{k_{pf}}$; otherwise augment B by the arc (r,p) for which $w_r = k_{rp}w_p - c_{rp}$. In either case, if there is more than one arc then choose any such arc. Finally, if T(p) contains all the nodes the algorithm is completed; otherwise set $T(p+1) = T(p)$, $p = p+1$, and return to step 2.

4. Validity

In order to establish the validity of the algorithm, it is necessary to first prove the following lemma:

Lemma: At each iteration p,

$$\min_{i \notin T(p)}\{w_i^1(p)\} \geq \max_{i \notin T(p)}\{w_i^2(p)\}$$

(i.e., if w_f and w_r are both defined at iteration p, then $w_f \geq w_r$).

Proof:

The result is established by induction. By the lemma in [13], we can assume w.l.o.g. that all $w_i^1(p)$ and $w_i^2(p)$ at every stage p are non-negative if the first dual variable is set equal to $\overline{M}$. Additionally, we assume that the nodes have been renumbered in theorder in which their w_i's are fixed.

(i) For the first iteration (p=1), we may assume w.l.o.g. that w_f and w_r are defined and have the values $w_f = \dfrac{\overline{M} + c_{1f}}{k_{1f}}$ and $w_r = k_{r1}\overline{M} - c_{r1}$ Since $c_{ij} \geq 0$ for all $(i,j) \in N$ and $0 < k_{ij} \leq 1$ for all $(i,j) \in N$, $w_f \geq w_r$ and thus

$$\min_{i \notin T(1)}\{w_i^1(p)\} \geq \max_{i \notin T(1)}\{w_i^2(p)\}.$$

(ii) Now assume the hypothesis is true for the p^{th} stage. Thus

$$\min_{i \notin T(p)}\{w_i^1(p)\} \geq w_p \geq \max_{i \notin T(p)}\{w_i^2(p)\} \qquad (4)$$

and for each $i \notin T(p)$

$$w_i^1(p+1) = \begin{cases} \min\{w_i^1(p), \dfrac{w_p + c_{pi}}{k_{pi}}\} & \text{if } (p,i) \in N \\ w_i^1(p) & \text{if } (p,i) \notin N \end{cases}$$

and

$$w_i^2(p+1) = \begin{cases} \max\{w_i^2(p), k_{ip}w_p - c_{ip}\} & \text{if } (i,p) \in N \\ w_i^2(p) & \text{if } (i,p) \notin N \end{cases}$$

We may assume w.l.o.g. that for each $i \notin T(p)$

$$w_i^1(p+1) = \min\{w_i^1(p), \frac{w_p + c_{pi}}{k_{pi}}\}$$

and

$$w_i^2(p+1) = \max\{w_i^2(p), k_{ip}w_p - c_{ip}\}.$$

Therefore,

$$\min_{i \notin T(p+1)} \{w_i^1(p+1)\} = \min_{i \notin T(p+1)} \{ w_i^1(p), \frac{w_p + c_{pi}}{k_{pi}} \}$$

and

$$\max_{i \notin T(p+1)} \{w_i^2(p+1)\} = \max_{i \notin T(p+1)} \{w_i^2(p), k_{ip}w_p - c_{ip}\}.$$

To establish that $\min_{i \notin T(p+i)} \{w_i^1 (p + 1)\} \geq \max_{i \in T(p+i)} \{ w_i^2 (p+1)\}$,

it suffices to show that the following cases are true:

(a) $\min_{i \notin T(p+1)} \{w_i^1(p)\} \geq \max_{i \notin T(p+1)} \{w_i^2(p)\}$

(b) $\min_{i \notin T(p+1)} \{ \frac{w_p + c_{pi}}{k_{pi}} \} \geq \max_{i \notin T(p+1)} \{k_{ip}w_p - c_{ip}\}$

(c) $\min_{i \notin T(p+1)} \{w_i^1(p)\} \geq \max_{i \notin T(p+1)} \{k_{ip}w_p - c_{ip}\}$

(d) $\min_{i \notin T(p+1)} \{\frac{w_p + c_{pi}}{k_{pi}}\} \geq \max_{i \notin T(p+1)} \{w_i^2(p)\}$.

Case (a) follows directly from the induction hypothesis since

$$\max_{i \notin T(p+1)} \{w_i^2(p)\} \leq \max_{i \notin T(p)} \{w_i^2 (p)\}$$

and

$$\min_{i \notin T(p+1)} \{w_i^2(p)\} \geq \min_{i \notin T(p)} \{w_i^2 (p)\}.$$

Since, by the lemma in [13], $w_p \geq 0$, and $c_{ij} \geq 0$ and $0 < k_{ij} \leq 1$ for $(i,j) \in N$, case (b) is clearly true. Cases (c) and (d) follow from expression (4), $c_{ij} \geq 0$ for $(i,j) \in N$, and $0 < k_{ij} \leq 1$ for $(i,j) \in N$. Therefore, the lemma follows by induction.

<u>Theorem:</u> The algorithm obtains a feasible basis for Problem (2) or determines that the network associated with Problem (2) is disconnected.

Proof: The algorithm terminates after at most m iterations, because at each iteration T(p) is augmented by one node or else the algorithm stops with the assertion that a feasible basis has been obtained or that the network is disconnected.

The assertion that the network is disconnected occurs when S and R are empty and there exists some node $j \notin T(p)$. Under these conditions the network is split into at least two disconnected parts.

Ruling out disconnectedness, it must be demonstrated that the values assigned to the w_i are feasible, i.e., $k_{ij}\, w_j - w_i \leq c_{ij}$ for all $(i,j) \in N$. Again the method of proof is induction. Assume the nodes are numbered in the order in which they are fixed. At iteration r, the number of nodes in T(r) is r. Suppose that the w_j values for these nodes represent a feasible solution to the dual subproblem obtained by considering all nodes in T(r) and all links joining the nodes in T(r). We shall show that the process of augmenting the $(r+1)^{st}$ node to T(r) assigns to its dual variable w_{r+1} a value that is feasible for the dual subproblem obtained by considering the (r+1) nodes in T(r+1), thereby yeilding the desired inductive result.

(i) To begin , let r = 1. The assignment $w_1 = \overline{M}$ clearly provides a feasible solution to the subproblem consisting of the first node. For r = 2, the possibilities are that $w_2 = \dfrac{\overline{M} + c_{12}}{k_{12}}$ or $w_2 = k_{21}\, \overline{M} - c_{21}$.

If $w_2 = \dfrac{\overline{M} + c_{12}}{k_{12}}$, then $w_2 \geq w_1$ since $0 < k_{ij} \leq 1$ and $c_{ij} \geq 0$

for $(i,j) \in N$. Thus, the values of w_1 and w_2 satisfy the feasibility conditions not only for the arc (1,2), but also for the arc (2,1) if it exists. On the other hand, if $w_2 = k_{21}\, \overline{M} - c_{21}$ then $w_2 \leq w_1$. Therefore, the values of w_1 and w_2 satisfy the feasibility conditions not only for the arc (2,1) but also for the arc (1,2) if it exists.

(ii) Now assume that the r nodes in T(r) have node values which yield a dual feasible solution. It will be shown that adding the $(r+1)^{st}$ node then yields a dual feasible solution.

Let $p(r+1) = \{j \; ; \; j \in T(r),\ (j,\ r+1) \in N\}$ and
let $q(r+1) = \{j \; ; \; j \in T(r),\ (r+1,\ j) \in N\}$.

Suppose that node r+1 is being added in the "forward direction" (i.e., $w_{r+1} = \min\limits_{j \in p(r+1)} \{\dfrac{w_j + c_{jr+1}}{k_{jr+1}}\}$).

Then automatically w_{r+1} satisfies the condition $k_{j,r+1}\, w_{r+1} - w_j \leq c_{j,r+1}$ for $j \in p(r+1)$. By the previous lemma, $w_{r+1} \geq \max\limits_{j \in q(r+1)} \{k_{r+1,j} w_j - c_{r+1,j}\}$ and therefore the condition $k_{r+1,j} w_j - w_{r+1} \leq c_{r+1,j}$ for $j \in q(r+1)$ is satisfied.

Likewise, suppose node r+1 is being added in the "reverse direction" (i.e. $w_{r+1} = \max\limits_{j \in q(r+1)} \{k_{r+1,j} w_j - c_{r+1,j}\}$). Then the condition $k_{r+1,j} w_j - w_{r+1} \leq c_{r+1,j}$ for $j \in q(r+1)$ is automatically satisfied. Again by the lemma $w_{r+1} \leq \min\limits_{j \in p(r+1)} \{\frac{w_j + c_{j,r+1}}{k_{j,r+1}}\}$, the condition $k_{j,r+1} w_{r+1} - w_j \leq c_{j,r+1}$ for $j \in p(r+1)$ is satisfied. This completes the feasibility proof.

The final step of the proof is to show that the feasible solution thus determined by the algorithm is in fact basic feasible, i.e., the arcs in B form a basis when the algorithm terminates. First, upon termination B contains m-1 arcs because one new arc is added to B each time a node is added to T(p), except at initialization. The arcs in B clearly span the network. Further, the arcs in B contain no cycles. To verify this, assume the contrary and let (r,s) be the first arc added to B that creates a cycle with the previous arcs. Then there must be an arc $(r,j) \in B$ and an arc $(i,s) \in B$. But this is impossible when (r,s) is added to B, since all arcs $(i,j) \in B$ satisfy $i \in T(p)$ and $j \in T(p)$, whereas to augment B with the arc (r,s) either $r \notin T(p)$ or $s \notin T(p)$. This completes the proof.

5. Computational Results

To test the advanced start procedure, the dual codes developed in [9, 12] were modified to use this start. These codes use the double pricing procedure [12,14] for determining the unique updated linear equation which expresses the basic variable, leaving the basis in a pivot, as a linear combination of the current nonbasic variables (for a complete description see [12,14]). The important aspect is that the double pricing procedure is quite efficient for calculating these coefficients. Thus these codes should be efficient in the pivot phase of the simplex dual method [20].

All of the codes used in this study are in-core codes; i.e., the program and all of the problem data simultaneously reside in fast-access memory. They are all coded in FORTRAN and none of them have been tuned for a particular compiler. All of the problems were solved on the CDC 6600 at the University of Texas Computation Center using the Run compiler. The computer jobs were executed during periods when the machine load was approximately the same, and all solution times are exclusive of input and output, i.e., the total time spent solving the problem was recorded by calling a Real Time Clock upon starting to solve the problem and again when the solution was obtained.

Initially, the advanced dual start code was tested on eighteen of the benchmarked problems in [19] since these problems had already been solved using the codes by Bennington [2], Boeing, General Motors, SHARE [5,21], SUPERK [1], Texas Water Development Board, PNET-I [9], and DNET [9]. The first five of these codes are variants of the out-of-kilter method [7,8] except for Bennington [2]. PNET-I [9] is the fastest known special purpose primal simplex code for solving capacitated and uncapacitated transshipment problems. DNET is a special purpose dual simplex code using the start procedure in [13] for solving transshipment problems. DNET was modified to use the advanced dual start and this version will be referred to, henceforth as DNET-I.

Additionally, these 18 problems were chosen since they are generally available to other researchers and the problem set is reasonably small. The specifications of these 18 problems as required on the input cards to the network generator in [19] are given in Table I. Problems 1-5 are 100 x 100 transportation problems, problems 6-10 are 150 x 150 transportation problems, problems 11-15 are 200 x 200 assignment problems, and problems 16-18 are 1000 node transshipment problems. Table II contains the solution

times for each of these problems for each of the codes.

Comparing the advanced start dual code DNET-I against the previous dual code DNET indicates that our initial premise (i.e., since the majority of the computational time spent in the dual procedure is devoted to the selection of an in-coming arc, a considerable amount of time could be devoted to finding the starting basis, if the start provided a substantial reduction in the number of pivots to reach optimality) is correct. In particular, the results show that the difference in start times for the two codes is negligible and in some cases (surprisingly) the advanced start takes less time to find the starting basis. In terms of pivot reduction, the advanced start is quite extraordinary. For example at least a 14% reduction is achieved on every problem and in five cases, the reduction is more than 50%. This pivot reduction is directly reflected in the total solution time since DNET-I strictly dominated DNET in each case and the percent reduction in total solution time is almost identical to the percent pivot reduction. Unfortunately, this substantial reduction is not enough to make the dual method competitive with PNET-I or SUPERK on this broad class of problems. However DNET-I is comparable with the other codes Bennington, SHARE, Boeing, General Motors, and Texas Water Development Board. Another noteworthy feature of the computational results is that PNET-I strictly dominates all of the codes. In fact PNET-I is roughly twice as fast as SUPERK, the fastest known out-of-kilter code. This result is extremely important since it completely contradicts the folklore that the out-of-kilter method is the fastest (in terms of total solution time) means for solving transportation and transshipment problems.

Motivated by Harris' [15] hypothesis that the dual simplex method might be an effective solution approach for solving extremely rectangular transportation problems (i.e., problems which have only a few origins and many destinations), we decided to solve some rectangular problems using the fastest code of each type. In order to make our research results available to fellow researchers, we used NETGEN [19] to generate 12 highly rectangular transportation problems. Table III contains the specifications for these problems as required on the input cards for NETGEN. The computational results on these problems are given in Table IV.

Important results which can be gleaned from Table IV are that DNET-I takes less time than SUPERK in all but one of these problems; however, DNET-I again assumes its secondary role to PNET-I, being slower on

all of the problems. Even though DNET-I is slower than PNET-I on these rectangular problems the magnitude of the differences is quite small compared with the Table II results. Thus Harris' hypothesis appears to have been well founded.

Unlike the square problems of Table II in which start selection time is a fairly small proportion of the total computation time, rectangular problems cause the dual to spend a great deal of time picking the start. For example, the problems where DNET-I performed best, in particular, problems 1,5,6,7, and 11, the start time constituted 94%, 82%, 71%, 76%, and 79% respectively of the total computation time. This appears to reinforce our original premise that quite a bit of time could be devoted to start selection if that start reduces the number of pivots being performed.

More research is needed in the area of determining precise boundaries within which DNET-I performs best. Initial results indicate that it performs better as the number of sources decreases, but there are also trends indicating that arc density plays an important role. Problems 8 and 9 show that as the arc density increases from 50% to 90% the number of pivots required to reach an optimal solution nearly doubles. Yet in problems 3 and 4, an increase from roughly 55% to roughly 75% caused nearly a 30% reduction in the number of pivots. In contrast, PNET-I is not as sensitive to arc density ranges, since the number of pivots is proportional to the number of arcs.

TABLE I

PROBLEM SPECIFICATIONS*

	Number of Nodes	Number of Sources	Number of Sinks	Number of Arcs	Cost Range Min	Cost Range Max	Total Supply	Random Number Seed	Objective Function Value
1	200	100	100	1300	1	100	100,000	13502460	2,153,303
2	200	100	100	1500	1	100	100,000	13502460	1,950,881
3	200	100	100	2000	1	100	100,000	13502460	1,565,928
4	200	100	100	2200	1	100	100,000	13502460	1,462,732
5	200	100	100	2900	1	100	100,000	13502460	1,342,058
6	300	150	150	3150	1	100	150,000	13502460	2,302,477
7	300	150	150	4500	1	100	150,000	13502460	2,047,034
8	300	150	150	5155	1	100	150,000	13502460	2,155,354
9	300	150	150	6075	1	100	150,000	13502460	1,775,454
10	300	150	150	6300	1	100	150,000	13502460	2,145,687
11	400	200	200	1500	1	100	200	13502460	4,563
12	400	200	200	2250	1	100	200	13502460	3,389
13	400	200	200	3000	1	100	200	13502460	3,070
14	400	200	200	3750	1	100	200	13502460	2,754
15	400	200	200	4500	1	100	200	13502460	2,721
16	1000	50	50	2900	1	100	1,000,000	13502460	122,582,531
17	1000	50	50	3400	1	100	1,000,000	13502460	105,050,119
18	1000	50	50	4400	1	100	1,000,000	13502460	86,331,458

TABLE III

	Number of Nodes	Number of Sources	Number of Sinks	Number of Arcs	Cost Range Min	Cost Range Max	Total Supply	Random Number Seed	Objective Function Value
1	100	2	98	150	1	100	30,000	13502460	1,316,643
2	200	5	195	500	1	100	30,000	13502460	914,078
3	500	4	496	1000	1	100	30,000	13502460	946,027
4	500	4	496	1500	1	100	30,000	13502460	743,330
5	1000	2	998	1500	1	100	30,000	13502460	1,193,805
6	1000	2	998	1800	1	100	30,000	13502460	1,127,833
7	1000	3	997	3000	1	100	30,000	13502460	730,224
8	1000	5	995	2000	1	100	30,000	13502460	934,100
9	1000	5	995	4500	1	100	30,000	13502460	594,816
10	1450	5	1445	2000	1	100	30,000	13502460	1,084,372
11	1500	2	1498	2500	1	100	30,000	13502460	1,154,911
12	1500	5	1495	6000	1	100	30,000	13502460	[illegible]

*All values not indicated are zero

TABLE II

Solution Times (In Seconds)

PROBLEM	DNET			DNET - I			PNET-I	SUPERK	SHARE	BENN	GM	BOEING	TWB
	Total Time	Start Time	No. of Pivots	Total Time	Start Time	No. of Pivots							
1	12.86	.59	535	7.61	.85	318	1.17	5.68	17.76	20.25	46.25	30.25	21.23
2	13.58	.63	532	10.87	.85	447	1.35	6.47	21.34	24.36	63.30	21.59	21.39
3	21.44	.75	757	12.64	.90	443	1.74	6.87	26.16	34.56	105.72	31.47	28.03
4	17.96	.81	577	11.98	.92	401	1.47	6.57	25.13	31.45	70.74	36.47	27.60
5	23.34	1.02	670	18.64	.97	546	1.73	6.77	30.97	52.10	90.10	47.73	NA
6	46.10	1.33	1188	25.81	1.[illegible]6	701	3.06	11.05	46.40	61.00	92.32	46.64	NA
7	74.88	1.77	1500	36.53	1.34	760	3.57	12.86	65.92	DNR	157.31	113.12	NA
8	97.93	2.03	1862	42.37	1.89	881	4.20	13.69	81.00	DNR	160.71	175.10	NA
9	101.65	2.42	1652	46.06	1.92	819	4.35	13.40	81.21	DNR	158.01	186.99	NA
10	95.96	2.52	1528	72.52	2.00	1318	5.57	14.13	84.24	DNR	197.82	184.75	NA
11	19.87	1.37	655	10.76	2.43	276	3.12	6.44	19.93	17.44	35.67	30.39	NA
12	26.58	1.51	770	16.00	2.54	274	[illegible].48	6.47	21.17	20.31	28.43	22.06	NA
13	42.38	1.70	1043	17.97	2.62	383	4.91	7.25	25.81	24.92	31.39	20.02	NA
14	NA	NA	NA	28.10	2.64	554	5.56	6.95	24.95	27.40	18.62	23.11	NA
15	NA	NA	NA	38.60	2.71	749	5.91	7.56	27.05	DNR	23.48	21.08	NA
16	NA	NA	NA	79.54	12.43	1247	5.67	13.91	53.87	DNR	DNR	83.98	NA
17	NA	NA	NA	88.50	12.65	1389	6.55	14.51	52.55	DNR	DNR	117.83	NA
18	NA	NA	NA	100.53	12.98	1385	8.10	16.00	61.33	DNR	DNR	152.21	NA

NA - NOT AVAILABLE

DNR - DID NOT RUN

TABLES IV

Solution Times (in seconds)

Problem	DNET - I Time	DNET - I # Pivots	PNET - I Time	PNET - I # Pivots	SUPERK Time
1	.292	1	.079	18	.882
2	1.960	58	.447	84	3.397
3	10.249	182	2.414	203	13.330
4	9.656	122	2.782	139	12.424
5	18.996	43	3.927	129	35.287
6	22.214	88	4.557	128	36.721
7	21.866	65	2.734	60	37.547
8	32.717	261	5.852	259	39.104
9	56.926	493	10.314	338	43.726
10	58.106	313	8.286	318	67.459
11	43.735	90	5.553	114	71.042
12	124.822	758	18.024	424	DNR

DNR - DID NOT RUN

References

[1] Barr, R. S., F. Glover, and D. Klingman, "An Improved Version of the Out-of-Kilter Method and a Comparative Study of Computer Codes." To appear in Mathematical Programming.

[2] Bennington, G. E. "An Efficient Minimal Cost Flow Algorithm," O. R. Report 75, North Carolina State University, Raleigh, North Carolina, June 1972.

[3] Charnes, A. and W. W. Cooper, Management Models and Industrial Applications of Linear Programming. Vols. I and II, Wiley, New York, 1961.

[4] Charnes, A. and W. M. Raike, "One-Pass Algorithms for Some Generalized Network Problems," Operations Research, 14(1966), 914-924.

[5] Clasen, R. J., "The Numerical Solution of Network Problems Using the Out-of-Kilter Algorithm," RAND Corporation Memorandum RM-5456-PR, Santa Monica, California, March, 1968.

[6] Ford, L. and D. Fulkerson, "A Primal-Dual Algorithm for the Capacitated Hitchcock Problem," Naval Research Logistics Quarterly, 4(1957), 47-54.

[7] Ford, L. and D. Fulkerson, Flows in Networks, Princeton University Press, Princeton, N. J., 1962.

[8] Fulkerson, D., "An Out-of-Kilter Method for Minimal Cost Flow Problems," J. SIAM, 9(1961), 18-27.

[9] Glover, F., D. Karney, and D. Klingman, "Implementation and Computational Comparisons of Primal, Dual and Primal-Dual Computer Codes for Minimum Cost Network Flow Problems," To appear in Networks.

[10] Glover, F., D. Karney, and D. Klingman, "The Augmented Predecessor Index Method for Locating Stepping Stone Paths and Assigning Dual Prices in Distribution Problems," Transportation Science, 6, 1(1972), 171-180.

[11] Glover, F., D. Karney, D. Klingman, and A. Napier, " A Computational Study on Start Procedures, Basis Change Criteria, and Solution Algorithms for Transportation Problems," Management Science, Vol. 20, 5(1974), 793-813.

[12] Glover, F. and D. Klingman, "Double-Pricing Dual and Feasible Start Algorithms for the Capacitated Transportation (Distribution) Problem," University of Texas at Austin, 1970.

[13] Glover, F., D. Klingman, and A. Napier, "Basic Dual Feasible Solutions for a Class of Capacitated Generalized Networks," Operations Research, 20, 1(1972), 126-137.

[14] Glover, F., D. Klingman, and A. Napier, "An Efficient Dual Approach to Network Problems," OPSEARCH, 9, 1(1972), 1-18.

[15] Harris, B., "Three Algorithms for the Transportation Problem," Working Paper, University of Pennsylvania.

[16] Jewell, W. S., "Optimal Flow through Networks with Gains," Operations Research, 10(1962), 476-499.

[17] Karney, D. and D. Klingman, "Implementation and Computational Study on an In-Core Out-of-Core Primal Network Code," Research Report CS 158, Center for Cybernetic Studies, University of Texas at Austin (1973).

[18] Klingman, D., A. Napier, and G. T. Ross, "A Computational Study of the Effects of Problem Dimensions on Solution Times for Transportation Problems," Research Report CS 135, Center for Cybernetic Studies, University of Texas at Austin (1973).

[19] Klingman, D., A. Napier, and J. Stutz, "NETGEN - A Program for Generating Large Scale (Un)Capacitated Assignment, Transportation, and Minimum Cost Flow Network Problems," Management Science, 20, 5(1974), 814-821.

[20] Lemke, C., "The Dual Method of Solving Linear Programming Problems," Naval Research Logistics Quarterly, 1(1954), 36-47.

[21] "Out-of-Kilter Network Routine," SHARE Distribution 3536, SHARE Distribution Agency, Hawthorne, New York, 1967.

[22] Wagner, H. M., Principles of Operations Research, Prentice-Hall, Englewood Cliffs, N. J., 1969.